IN THE CLAIMS:

This listing of claims will replace all prior revisions, and listings, of the claims in the application:

Listing of Claims:

1. (Currently Amended) A branch prediction architecture comprising:

a prediction selector;

a bimodal predictor coupled to the prediction selector, the bimodal predictor to generate a bimodal prediction for a branch instruction; ~~and~~

a plurality of global predictors coupled to the prediction selector, each global predictor to generate a corresponding global prediction for the branch instruction, the prediction selector to select a branch prediction from the bimodal prediction and the global predictions, wherein each global prediction is to be generated based on a different amount of global branch history information;

a first global predictor to generate a first global prediction by indexing into a first global array based on a first index, the first index to be associated with a first amount of global branch history information; and

a second global predictor to generate a second global prediction by indexing into a second global array based on a second index, the second index to be associated with a second amount of global branch history information and folding the second index to obtain a smaller index for use in indexing into the second global array, the first amount to be less than the second amount.

2. (Canceled)

3. (Canceled)

4. (Currently Amended) The branch prediction architecture of claim [[3]] 1, wherein the branch prediction architecture is to generate the first index by shifting a most recent branch bit into a

previous first stew to obtain a current first stew and performing an exclusive OR operation between the current first stew and one or more portions of an instruction address associated with the branch instruction, and to generate the second index by shifting the most recent branch bit into a previous second stew to obtain a current second stew and performing an exclusive OR operation between the current second stew and one or more portions of the instruction address, the previous and current first stews to have a length that corresponds to the first amount and the previous and current second stews to have a length that corresponds to the second amount.

5. (Currently Amended) The branch prediction architecture of claim [[3]] 1, wherein the branch prediction architecture is to generate the first index by shifting a most recent branch bit into a previous first global branch history to obtain a current first global branch history and performing an exclusive OR operation between the current first global branch history and one or more portions of an instruction address associated with the branch instruction, and to generate the second index by shifting the most recent branch bit into a previous second global branch history to obtain a current second global branch history and performing an exclusive OR operation between the current second global branch history and one or more portions of the instruction address, the previous and current first global branch histories to have a length that corresponds to the first amount and the previous and current second global branch histories to have a length that corresponds to the second amount.

6. (Currently Amended) The branch prediction architecture of claim [[3]] 1, wherein the plurality of global predictors includes a third global predictor to generate a third global prediction by indexing into a third global array based on a third index, the third index to be associated with a third amount of global branch history length, the second amount being less than the third amount.

7. (Original) The branch prediction architecture of claim 1, wherein the prediction selector includes:

a first multiplexer to generate an intermediate prediction based on the bimodal prediction, a first global prediction and whether a hit has occurred in a first global array; and

a second multiplexer to select the branch prediction based on the intermediate prediction, a second global prediction and whether a hit has occurred in a second global array.

8. (Original) The branch prediction architecture of claim 7, wherein the second multiplexer is to select the second global prediction if a hit notification is received from the second global array and select the intermediate prediction if a hit notification is not received from the second global array.

9. (Original) The branch prediction architecture of claim 8, wherein the first multiplexer is to select the first global prediction if a hit notification is received from the first global array and select the bimodal prediction if a hit notification is not received from the first global array.

10. (Original) The branch prediction architecture of claim 7, further including allocation logic coupled to the prediction selector, the allocation logic to allocate an entry in the first global array to the branch instruction if the branch prediction results in a misprediction and originates from the bimodal predictor.

11. (Original) The branch prediction architecture of claim 10, wherein the allocation logic is to allocate an entry in the second global array to the branch instruction if the branch prediction results in a misprediction originated from a first global predictor, where the first global predictor generates the first global prediction.

12. (Original) The branch prediction architecture of claim 7, further including update logic coupled to the predictors, the update logic to update a bimodal array of the bimodal predictor based on an actual branch outcome associated with the branch prediction.

13. (Original) The branch prediction architecture of claim 12, wherein the update logic is to update the second global array based on the actual branch outcome if the tag of the branch instruction matched a tag in the second global array.

14. (Original) The branch prediction architecture of claim 12, wherein the update logic is to update the first global array based on the actual branch outcome if the tag of the branch instruction

matched a tag in the first global array.

15. (Original) The branch prediction architecture of claim 12, wherein the update logic is to update the first global array based on the actual branch outcome if the tag of the branch instruction did not match a tag in the second global array and if the tag of the branch instruction matched a tag in the first global array.

16. (Original) The branch prediction architecture of claim 1, wherein the branch prediction is to include a predicted direction of the branch instruction.

17. (Original) The branch prediction architecture of claim 16, wherein the branch prediction is to further include an instruction target address of the branch instruction.

18. (Currently Amended) A branch prediction architecture comprising:

a prediction selector having a first multiplexer and a second multiplexer;

a bimodal predictor coupled to the prediction selector, the bimodal predictor to generate a bimodal prediction for a branch instruction;

a plurality of global predictors coupled to the prediction selector, each global predictor to generate a corresponding global prediction for the branch prediction, the first multiplexer to generate an intermediate prediction based on the bimodal prediction, a first global prediction and whether a hit has occurred in a first global array, the second multiplexer to select a branch prediction based on the intermediate prediction, a second global prediction and whether a hit has occurred in a second global array;

allocation logic coupled to the prediction selector, the allocation logic to allocate an entry in the first global array to the branch instruction if the branch prediction results in a misprediction and originated from the bimodal predictor, the allocation logic to allocate an entry in the second global array to the branch instruction if the branch prediction results in a misprediction and originated from the first global prediction; ~~and~~

update logic coupled to the predictors, the update logic to update a bimodal array of the bimodal predictor based on an actual branch outcome associated with the branch prediction, each

global prediction to be generated based on a different amount of global branch history information;

a first global predictor to generate a first global prediction by indexing into a first global array based on a first index, the first index to be associated with a first amount of global branch history information; and

a second global predictor to generate a second global prediction by indexing into a second global array based on a second index the second index to be associated with a second amount of global branch history information and folding the second index to obtain a smaller index for use in indexing into the second global array, the first amount to be less than the second amount, the first history amount to be less than the second amount.

19. (Canceled)

20. (Currently Amended) The architecture of claim [[19]] 18, wherein the branch prediction architecture is to generate the first index by shifting a most recent branch bit into a previous first stew to obtain a current first stew and performing an exclusive OR operation between the current first stew and one or more portions of an instruction address associated with the branch instruction, and to generate the second index by shifting the most recent branch bit into a previous second stew to obtain a current second stew and performing an exclusive OR operation between the current second stew and one or more portions of the instruction address, the previous and current first stews to have a length that corresponds to the first amount and the previous and current second stews to have a length that corresponds to the second amount.

21. (Currently Amended) The architecture of claim [[19]] 18, wherein the branch prediction architecture is to generate the first index by shifting a most recent branch bit into a previous first global branch history to obtain a current first global branch history and performing an exclusive OR operation between the current first global branch history and one or more portions of an instruction address associated with the branch instruction, and to generate the second index by shifting the most recent branch bit into a previous second global branch history to obtain a current second global branch history and performing an exclusive OR operation between the current second global branch history and one or more portions of the instruction address, the previous and current first global branch

histories to have a length that corresponds to the first amount and the previous and current second global branch histories to have a length that corresponds to the second amount.

22. (Original) The architecture of claim 18, wherein the branch prediction is to include a predicted direction of the branch instruction.

23. (Original) The architecture of claim 22, wherein the branch prediction is to further include an instruction target address of the branch instruction.

24. (Canceled)

25. (Canceled)

26. (Canceled)

27. (Currently Amended) A method of processing a branch instruction comprising:

generating a bimodal prediction for the branch instruction;

generating a plurality of global predictions for the branch instruction; ~~and~~

selecting a branch prediction from the bimodal prediction and the global predictions, wherein each global prediction is generated based on a different amount of global branch history information

generating a first global prediction by indexing into a first global array based on a first index, the first index being associated with a first amount of global branch history; and

generating a second global prediction by indexing into a second global array based on a second index, the second index being associated with a second amount of global history and folding the second index to obtain a smaller index for use in indexing into the second global array, the second history amount being less than the first amount.

28. (Canceled)

29. (Canceled)

30. (Currently Amended) The method of claim [[29]] 27, further including:

generating the first index by shifting a most recent branch bit into a previous first stew to obtain a current first stew and performing an exclusive OR operation between the current first stew and one or more portions of an instruction address associated with the branch instruction; and

generating the second index by shifting the most recent branch bit into a previous second stew to obtain a current second stew and performing an exclusive OR operation between the current second stew and one or more portions of the instruction address, the previous and current first stews to have a length that corresponds to the first amount and the previous and current second stews to have a length that corresponds to the second amount.

31. (Currently Amended) The method of claim [[29]] 27, further including:

generating the first index by shifting a most recent branch bit into a previous first global branch history to obtain a current first global branch history and performing an exclusive OR operation between the current first global branch history and one or more portions of an instruction address associated with the branch instruction; and

generating the second index by shifting the most recent branch bit into a previous second global branch history to obtain a current second global branch history and performing an exclusive OR operation between the current second global branch history and one or more portions of the instruction address, the previous and current first global branch histories to have a length that corresponds to the first amount and the previous and current second global branch histories to have a length that corresponds to the second amount.

32. (Currently Amended) The method of claim [[29]] 27, further including generating an intermediate prediction based on the bimodal prediction, a first global prediction and whether a hit has occurred in a first global array, the branch prediction being selected based on the intermediate prediction, a second global prediction and whether a hit has occurred in a second global array.

33. (Original) The branch prediction architecture of claim 1, wherein each global prediction is generated at an instruction fetch stage of a pipeline.

34. (Canceled)